Of Sampling and Smoothing: Approximating Distributions over Linked Open Data

Thomas Gottron

Institute for Web Science and Technologies
University of Koblenz-Landau, Germany
gottron@uni-koblenz.de

Abstract Knowledge about the distribution of data provides the basis for various tasks in the context of Linked Open Data, e.g. for estimating the result set size of a query, for the purpose of statistical schema induction or for using information theoretic metrics to detect patterns. In this paper I investigate the potential of obtaining estimates for such distributions from samples of Linked Data. Therefore, I consider three sampling methods applicable to public RDF data on the Web as well as smoothing techniques to overcome the problem of unseen events in the sample space of a distribution. In a systematic empirical evaluation I look into the effects of these techniques on the quality of distributions approximated from samples. The obtained insights help to assess the quality impact of combinations of sampling and smoothing techniques on five prototypical structures over which distributions are estimated. Furthermore, the results demonstrate the potential and the limitations of these techniques, motivating further work in the direction of sampling from Linked Open Data.

1 Introduction

Estimating the distribution of instances in data storage systems plays an important role for various applications. Query optimisation techniques rely on such information for estimating the result set size for a query [17]. Data mining solutions perform statistical analytics to detect patterns in the data [7]. Encoding algorithms may make use of a distribution for an efficient compression and storage of the managed data and in evaluation settings distributions have been used to assess the stability of index structures over evolving data [6]. In the context of relational database management systems the analysis of stored data for the purpose of estimating distributions and densities has a long tradition [16,10,15]. For data published on the Linked Open Data (LOD) cloud, so far, such analytics have been pursued far less. Only in a few, very specific settings the distribution of data on the LOD cloud has been investigated.

One reason might be the lack of a predefined schema which provides a clear structure over which to estimate distributions of data items. Another reason might be the difficulty to obtain sufficient data for reliable estimates of a distribution. However, recent developments provide some techniques and solutions to these problems. For instance, methods for schema induction and schema-level indices [14,20] can be used to extract a schema on the basis of data observations. In fact, such index structures have successfully been employed in settings where densities were used for a large scale analysis of

LOD [8]. Thus, it seems the right time to consider a more thorough investigation of how to obtain reliable estimates of data distributions in an efficient and scalable way which is suitable for processing data on the Linked Data cloud.

With this paper I would like to take a first step in this direction. Thus, I address two of the challenges which will need to be solved for efficient and effective estimation of distributions over Linked Data: *sampling* and *smoothing*. Both will play a major role when dealing with distributions over Linked Data in the future. Sampling techniques will be a key contributing factor as they enable large scale processing of Linked Data. Given the recent growth rate of the Web of Data it is becoming less and less feasable to analyse the entirety of all data published on the LOD cloud. Moreover, the dynamics and change rate [4] of the data call for a large scale monitoring of the data to keep distributions up to date. The obvious response to counter this problem is to downscale the data volume—and here comes sampling into the play.

The second topic—smoothing—is closely related to sampling, but addresses another dimension of the problem. The liberty of data providers to model data as they please as well as the decentralised approach of everyone being able to add, change or remove data in an ad-hoc fashion will lead to a particular phenomenon: the sample space over which to model a distribution is not fixed but the events in this space are flexible and evolving. Take, for instance, the question of how data instances are distributed over RDF class types. At any time a data provider can come up with a new class type and start using it immediately to describe entities on the Web. Any distribution which has been obtained over previous data will not be aware of this class type. Accordingly any application making use of the distribution will operate with a zero probability. This causes problems in various contexts and applications (e.g. when performing an information theoretic analysis). The importance of this problem becomes even more obvious when introducing sampling approaches. Using only a small slice of the Linked Data cloud to estimate distributions over the entire data space will almost certainly mean that some rare combinations of data characteristics have not been observed in the sample. Smoothing techniques aim at overcoming this problem, by reserving some probability mass for unseen events.

Hence, given the importance of sampling and smoothing, in this paper I will consider three well established approaches for sampling Linked Data as well as smoothing techniques to overcome the problem of unseen events. The investigated distributions and their sample spaces of schema-level structures are taken from related work and represent typical information used in different application settings. The contribution of the paper at hand is therefore an analysis of the impact of different combinations of sampling and smoothing on estimating distributions over different types of schema structures. Furthermore, the paper provides a baseline for future work on more specific and sophisticated approaches for either sampling or smoothing techniques.

The remainder of the paper is structured as follows. I will formalise the data model used throughout the paper as well as the considered sampling techniques in Section 2. Subsequently, Section 3 gives an overview of schema-level structures which are frequently used as events in a sample space for estimating distributions over Linked Data and presents classical smoothing techniques. An empirical evaluation and comparison of combinations of smoothing and sampling techniques and their impact on the qual-

ity of the obtained distributions is presented in Sections 4 and 5. Finally, I will review related work in Section 6 before concluding the paper with a summary of the findings and a roadmap for future work.

2 Sampling Linked Open Data

Linked Open Data can be perceived as a distributed labeled graph. This view provides the basis for formalising the underlying data model in the N-Quad [3] representation. In this representation, Linked Data comes in the form of quads (s, p, o, c) where s corresponds to the subject of an RDF triple, p to the predicate and o to the object. The last entry c provides the context, i.e. the URI at which this triple has been published on the LOD cloud. Formally, a data set of quads is a relation $R \subset (U \cup B) \times U \times (U \cup B \cup L) \times U$, where U is the set of all (possible) URIs, B the set of blanks nodes, L the set of literals and U, B and L are pairwise disjunct.

The sampling approaches discussed in this paper can be formalised as a random selection process over a set $\mathcal{B}$ of base elements. This set $\mathcal{B}$ is derived from the quads R, but focusses on certain aspects of the data. All sampling approaches considered in this paper are then based on selecting elements in $\mathcal{B}$ according to a uniform probability distribution.

The set of elements selected from $\mathcal{B}$ then defines the criteria for selecting a subset of corresponding and relevant quads from the set R. Thus, we need two functions: (1) the function *restrict* : $R \rightarrow \mathcal{B}$ which provides a base element for a quad and (2) a function *expand* : $\mathcal{B} \rightarrow \mathcal{P}(R)$ which provides for each base element in $\mathcal{B}$ the set of quads relevant to it. Sampling is then performed over the set $\mathcal{B}$ using a uniform distribution, i.e. the probability of selecting a specific element $b \in \mathcal{B}$ is $p(b) = \frac{1}{|\mathcal{B}|}$. The overall sampling approach is implemented in three steps: (1) by computing the set $\mathcal{B}$ from R using *restrict*, (2) by performing a sampling on $\mathcal{B}$ and (3) by expanding the obtained subset of $\mathcal{B}$ into a reduced RDF data graph via the *expand* function.

There are three straight forward approaches for sampling which can easily be expressed in such a formalisation framework. None of them requires a deeper interpretation or analysis of the data or the incorporation of background knowledge. I will focus on these approaches because they are unbiased and therefore should be more suitable to provide a representative sample. (See Section 6 for related work on biased sampling techniques).

Triple (Edge) Based Sampling. The RDF data graph can be sampled using an edge based approach. This means to directly select the edges with a probability inverse proportional to the overall number of edges. The edges correspond to the triple statements of the elements in R, as each quad contains a description for the connection of two nodes. Formally, the functions *restrict* and *expand* are defined as follows.

$$\textit{restrict} : (s, p, o, c) \mapsto (s, p, o) \tag{1}$$

$$\textit{expand} : (s, p, o) \mapsto \{(s, p, o, c) \mid \exists c : (s, p, o, c) \in R\}. \tag{2}$$

USU (Node) Based Sampling. The second standard approach for sampling graph data is to select a subset of the nodes. The sample of the graph is then composed on the basis of the selected nodes and their adjacent edges. In RDF this corresponds to sampling from the set of URIs used to model entities. This means that literals and blank nodes can be ignored. In the context of this paper the sampling is implemented on the basis of URIs appearing in the subject positions of triples, so called *Unique Subject URIs* (USU)[1]. Once the USUs have been selected in the random process the sample of the Linked Data graph is constructed by adding all edges which are adjacent to a URI in this set. In the formal representation this leads to the following definitions for the two functions:

$$\textit{restrict} : (s, p, o, c) \mapsto s \tag{3}$$

$$\textit{expand} : s \mapsto \{(s, p, o, c) \mid \exists p, o, c : (s, p, o, c) \in R\}. \tag{4}$$

Context Based Sampling. The last sampling paradigm I will cover in this paper is based on the context of a quad. This means sampling is performed on the set of data sources on the Linked Data cloud. Once a data source is selected for inclusion in the sample, all data provided by this source is used to estimate a distribution. Effectively, in the functions *restrict* and *expand* we only need to consider the context c provided in the quads.

$$\textit{restrict} : (s, p, o, c) \mapsto c \tag{5}$$

$$\textit{expand} : c \mapsto \{(s, p, o, c) \mid \exists s, p, o : (s, p, o, c) \in R\}. \tag{6}$$

Context based sampling is probably the most natural approach for sampling on the LOD cloud. It aligns very well with the paradigm of dereferencing a URI to look up information provided there. Thus, selecting a subset of all URIs for dereferencing and using all information made available under these URIs is intuitive and easy to implement in practical applications. Nevertheless, we consider the other two sampling models for the sake of completeness and for providing a comparison.

3 Estimating and Smoothing Densities

For any density estimation it is necessary to first define a sample space Ω. It is the events in this space Ω for which we attempt to determine probabilities. In the context of Linked Data the events are defined on the basis of the observed triples and typically align with a structural or content related feature of the graph and its nodes and edges.

Related work provides several considerations about different kinds of data structures which can serve as events. For instance, a common scenario is to ask how likely

[1] Theoretically, an alternative would be to consider also URIs in the object position. However, given the structures which serve as events in the sample space this will have no practical effect on the distributions. The reason is, that an object URI which never appears in the subject position of a triple will not be assigned to any of these structures.

it is to observe a particular RDF class type or a particular predicate. Density information about this basic structures can serve in RDF triple stores to estimate the size of a result set. More complex events can be obtained when combining these basic structures into *property sets* (PS, also referred to as characteristic sets) [17,14], *type sets* (TS) [14] or *extended characteristic sets* (ECS) combining arbitrary sets of types and predicates [4,7].

The notation in this paper is based on the definition of Linked Data index models [6] and considers a set $\mathcal{K}$ of structural elements as the events in the sample space Ω. Thus, it is the elements in $\mathcal{K}$ for which we seek to estimate the distribution of data items D. A function $\sigma_D : \mathcal{K} \rightarrow \mathcal{P}(D)$ provides the set of data elements in D which comply with a given structure definition in $k \in \mathcal{K}$, e.g. all entities of a specific RDF type. Thus, σ_D provides us with a formal function to assign observations in the data to the events in the sample space Ω.

If we have access to the full data set D, we basically have observed the full population and all frequencies. Thus, we can easily estimate the distribution using a *maximum likelihood estimation* (MLE). In this case the event $k \in \Omega$ has a probability $\hat{P}_{MLE}(k) = \frac{|\sigma_D(k)|}{N}$, where $|\sigma_D(k)|$ is the number of observations we have made for k in D and N is the overall number of all observations.

When operating on a sample S of the full data set D, the estimation of the distribution of the events becomes more difficult for several reasons. First of all, certain events $k \in \Omega$ might not have been observed. Thus, it is difficult to construct Ω itself from the observations. A common approach in comprable scenarios (e.g. language modelling) is to introduce an artificial event <UNKNOWN>. This event is a representative for all events which were not foreseen in the sample space constructed over a sample S of data D.

However, using a maximum likelihood estimator would still assign a zero probability to this event, because the function σ_S operating on the sample S would provide an empty set. Furthermore, also the estimates for the seen events will be skewed. For instance, an event k which in the full data set occurred exactly once (i.e. $|\sigma_D(k)| = 1$) and which is contained in the random sample (i.e. also $|\sigma_S(k)| = 1$) would be overestimated in its probability, as the size of the sample is smaller then the overall population. Thus, if M is the number of events observed in the sample, a maximum likelihood estimation $\hat{P}_{MLE} = \frac{|\sigma_S(k)|}{M} = \frac{1}{M}$ is too high by a factor of $\frac{N}{M}$.

Smoothing techniques are meant to overcome such problems. They *smooth* the distribution by removing probability mass from the made observations and redistributing it to unseen events. This should counterbalance both of the above mentioned problems. The smoothing techniques used in this paper are very well established but also quite simplistic. Yet, they can serve as future baseline for more sophisticated approaches.

Laplace Smoothing In Laplace smoothing the number of observations for each event k is artificially increased by one (therefore it is sometimes also referred to as *add-one smoothing*). This gives an estimator of:

$$\hat{P}_{Laplace}(k) = \frac{|\sigma_S(k)| + 1}{M + |\Omega|} \tag{7}$$

While there are certain conceptual critics towards Laplace smoothing (e.g. a tendency to overestimate rare events), it is commonly applied in practice.

Lidstone Smoothing Lidstone Smoothing is a generalisation of Laplace smoothing. Instead of adding a value of one to all event counts, it involves a parameter λ:

$$\hat{P}_{Lidstone,\lambda}(k) = \frac{|\sigma_S(k)| + \lambda}{M + \lambda|\Omega|} \tag{8}$$

The smaller λ is chosen the closer is $\hat{P}_{Lidstone,\lambda}$ to a maximum likelihood estimator $\hat{P}_{MLE}$. For very high values of λ, $\hat{P}_{Lidstone,\lambda}$ gets closer to a uniform distribution.

4 Empirical Evaluation

The general idea of the empiric evaluation provided here is to see how distributions estimated over different sample sizes, sampling approaches and smoothing techniques deviate from the distribution over a full dataset.

4.1 Metrics

The task addressed in this paper is the comparison of distributions. In particular, we want to find out how close a distribution obtained from a sample is to the distribution over the full data set. A common metric to compare density functions and distributions is *Kullback-Leibler divergence*. Kullback-Leibler divergence compares two distributions in an information theoretic context and provides an asymmetric distance between the distributions. In this section I will briefly introduce the definition of this distance function and explain the interpretation in the context of compression theory.

Kullback-Leibler divergence is based on the definition of cross entropy. Assume we have two probability distributions $P_1(X)$ and $P_2(X)$. In the setting of this paper, P_1 would correspond to the true distribution over the full data while P_2 is the distribution estimated over a sample. Then the cross entropy is defined as:

$$H(P_1, P_2) = -\sum_{k \in \mathcal{K}} P_1(X = k) \log(P_2(X = k)) \tag{9}$$

In the context of compression theory, cross entropy can be interpreted as the average number of bits needed to encode events following the distribution P_1 based on an optimal prefix-free code [11] derived from P_2. If the two distributions are equivalent, then cross entropy corresponds to the normal entropy $H(P_1)$. The entropy of P_1 also provides a lower bound for cross entropy. Based on this interpretation, the Kullback-Leibler divergence gives the deviation in entropy relative to the entropy for P_1 and is defined as:

$$D_{KL}(P_1, P_2) = H(P_1, P_2) - H(P_1) \tag{10}$$

Therefore, if two distributions are equivalent, they have a Kullback-Leibler divergence of zero. This is a desirable feature for our evaluation as it renders the comparison of distributions over samples of data independent from the different levels of the entropy observed for different sample spaces defined by different observable structures.

Table 1. Average number of triples for sampling data based on the sampling method and the sampling rate.

Sampling-Rate	Sampling Method		
	Context	Triple	USU
0.05	825,276	817,050	808,085
0.1	1,645,758	1,638,891	1,645,041
0.2	3,308,450	3,279,957	3,294,814
0.3	4,903,447	4,917,632	4,916,383
0.4	6,487,714	6,547,144	6,546,638
0.5	8,134,887	8,194,817	8,181,503
0.6	9,895,754	9,828,583	9,840,084
0.7	11,487,162	11,462,177	11,451,802
0.8	13,151,521	13,099,584	13,099,766
0.9	14,765,467	14,740,145	14,767,641

4.2 Data Set and Experimental Setup

We use data from the Dynamic Linked Data Observatory (DyLDO) data set [13,12]. The motivation for using this data set is its careful design to cover a wide range of data source and many different domains. Under many aspects the data set has been designed to correspond to a representative extract of the Linked Data cloud. Details on the design considerations and their implementation for the data set can be found in the original publications. The DyLDO data set provides weekly crawls of LOD data sources starting from always the same set of seed URIs. However, as in the context of this paper we are not interested in temporal aspects but merely want to leverage the well motivated composition of DyLDO we only use the initial snapshot taken at the 6th of May 2012 which contains 16,376,867 RDF statements in N-Quad format.

Starting from this full data set I generated samples of decreasing size. The sample size was defined by a sampling rate. This rate covered values from 90% down to 10% in steps of 10% and an additional very small sample of 5%. To avoid unfortunate random configurations of the samples I repeated the process for each sample size and sampling method ten times. Thus, in total I created 300 samples (three sampling techniques, ten sample sizes, ten iterations), which were all taken independently from each other (in the probabilistic sense). Table 1 shows the average size (in number of triples) of the samples for each sampling method and sampling rate. The numbers illustrate that none of the sampling techniques exhibits a systematic tendency to generate too large or too small samples. This is visible in particular also in Figure 1 which depicts the same numbers as a plot. The lines for each sampling technique are essentially the same.

For each sample I constructed index structures[2] to obtain frequency counts and estimated distributions using the different smoothing techniques. For Lidstone smoothing the parameter was set to $\lambda = 0.5$, 0.1, and 0.01. For each such configuration of sam-

[2] The implementation of the index structures as well as sampling and smoothing techniques is available under an open source license at `https://github.com/gottron/lod-index-models`.

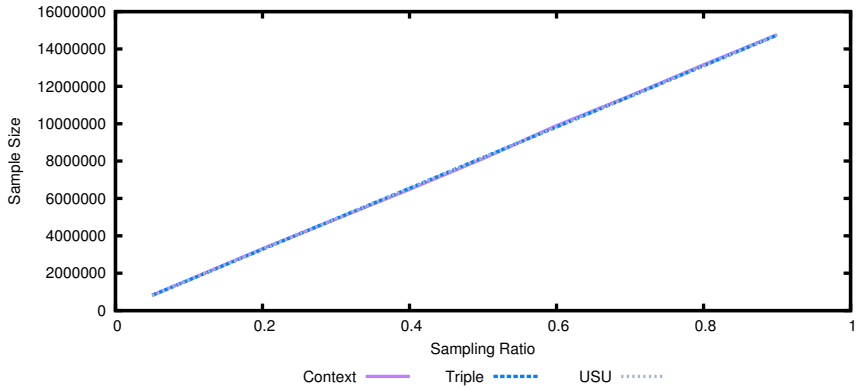


Figure 1. Average size of the samples for each of the sampling techniques depending on the sample rate.

pling technique and smoothing approach I then computed the distribution for each of the samples and compared it via Kullback-Leibler divergence to a maximum likelihood estimation of the distribution over the full dataset. The obtained values were used to plot a curve of how the divergence evolves on the basis of the sampling rate. This process is also illustrated in Figure 2. The plots presented in the subsequent section actually show the average values for Kullback-Leibler obtained over the ten samples generated for each combination of sampling technique and sampling rate.

5 Results

Let us first compare the different sampling techniques. In Figure 3 we can observe how Kullback-Leibler divergence increases when reducing the size of the sample. Each plot corresponds to a different structure used as events for the sample space (RDF types, predicates, type sets, predicate sets and extended characteristic sets). The different lines in the plot demonstrate the value of the Kullback-Leibler divergence for the three different sampling strategies based on triples, USUs and context. All distributions for the settings in Figure 3 were smoothed using the Laplace approach.

The plots illustrate nicely how the different sampling strategies affect the quality of the density estimations. Sampling based on USUs shows a very low Kullback-Leibler divergence even for high sampling rates. This behaviour is consistent across all types of structures considered as events. However, we can also observe that the quality of distributions estimated for simpler structures, e.g. the RDF types or predicates, is better than for the more complex structures, e.g. ECS. This is plausible as the definition of an extended characteristic set involves more triple statements, namely each RDF type assignment and all used predicate URIs. Thus, sampling is more likely to cause the loss of some of the information required to reliable obtain the actual structure.

This is also the reason why triple based sampling demonstrates the worst quality on this type of structures. When using triple based sampling for a subsequent estimation

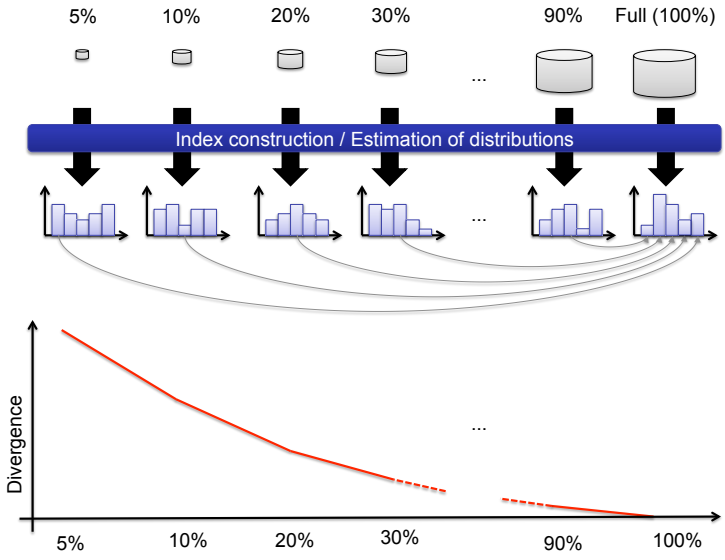
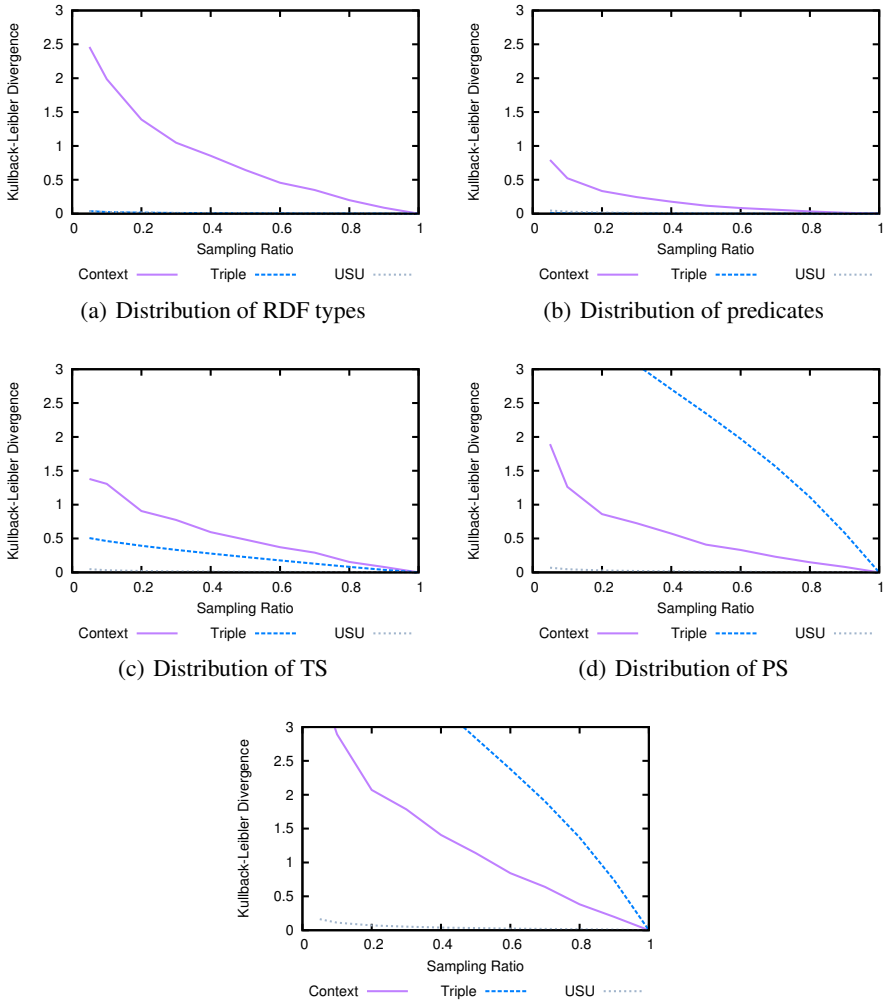


Figure 2. Evaluation process: On the basis of the full data set I took independent samples of different sizes using the different sampling techniques. Each of the samples was used with different smoothing techniques to estimate the distribution of the full data set.

of the distribution of PS and ECS structures Kullback-Leibler divergence is increasing very fast, even for high sampling rates of around 80%. Obviously, sampling just a fraction of the predicates leads to wrong structures and to strong deviations in the overall distribution. For type sets, interestingly, this effect is far less strong. One explanation can be that there are less type set combinations than property sets combinations in the analysed data set (25,727 TS vs. 35,985 PS). Thus, there are far less combinations which can be confused and distort the distribution. Furthermore, entities are typically annotated with only a few RDF types (if at all). Thus even after sampling, the odds are quite high to still observe a quite representative set of RDF types and their combinations. For the distributions over structures which are based on single triple statements, i.e. the RDF types and predicates, triple based sampling is behaving extremely well and comparable to or even slightly better than USU based sampling.

When employing sampling based on the context, the quality of estimated distribution is never the best. However, on the more complex structures it behaves better than the triple based sampling. And for sampling rates going down to 50% the Kullback-Leibler divergence also is not too high for all structures. An explanation for the sub-optimal behaviour is that context based sampling is prone to a domain bias. Based on the actual contexts sampled certain “regions” of the Linked Data cloud might be covered better than others. This might also explain that the plots of Kullback-Leibler divergence over context samples show more variance and look more rugged than the plots for the other sampling techniques.



(a) Distribution of RDF types

(b) Distribution of predicates

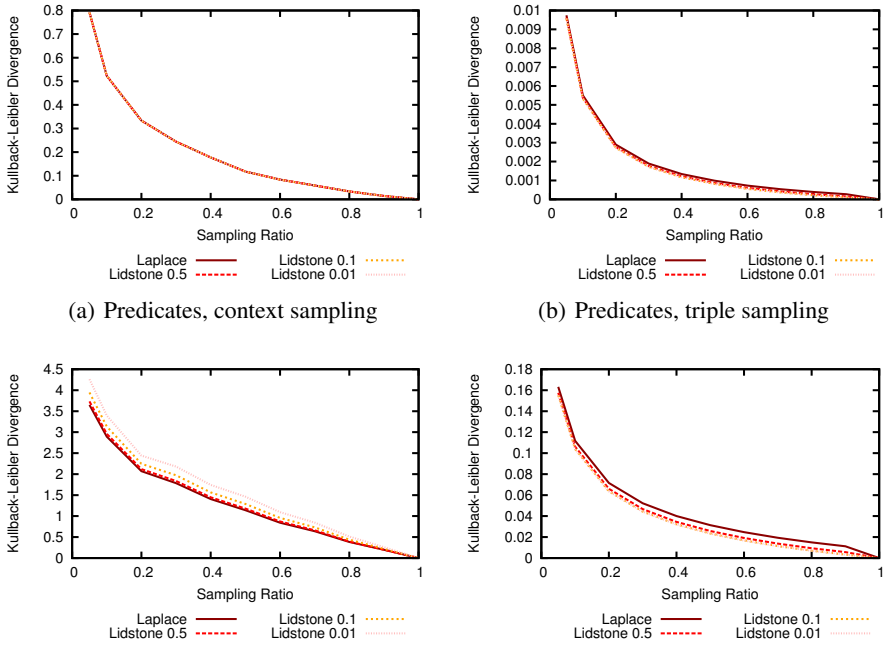
(c) Distribution of TS

(d) Distribution of PS

(e) Distribution of ECS

Figure 3. Comparison of Kullback-Leibler divergence for the three different sampling strategies when using Laplace smoothing.

In Figure 4 we see the impact of different smoothing techniques on the Kullback-Leibler divergence. The figures do not show all combinations, but illustrate some of the corner cases as well as realistic settings. Figure 4(a), for instance, shows how the distributions estimated over predicates behave when using a context based sampling. The choice of the smoothing model shows very little effect on the quality of the estimated distribution. Also in Figure 4(b), showing the divergence for triple based sampling, the lines for the different smoothing techniques essentially match. Note, the different reso-



(a) Predicates, context sampling

(b) Predicates, triple sampling

(c) ECS, context sampling

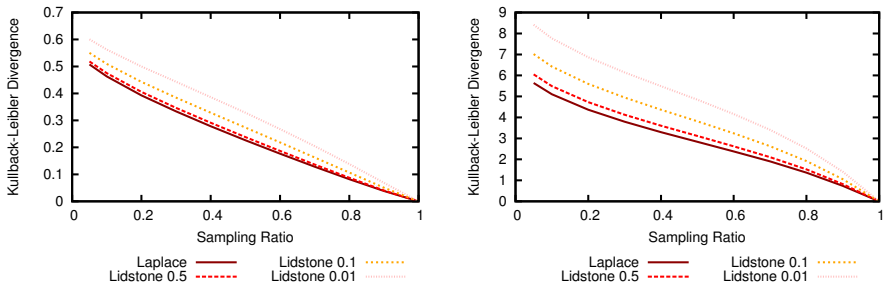
(d) ECS, USU sampling

Figure 4. Comparison of smoothing techniques for selected structure types and sampling methods.

lution on the y-axis. As stated above, the quality for predicate estimates is by far better when using triple based sampling. However, with the more detailed resolution in Figure 4(b), we can identify a slight advantage for Lidstone smoothing with a small value for parameter λ.

Looking at the estimates for the more challenging ECS structures, instead we observe a different behaviour. In Figure 4(c) we see again context based sampling. However, in this setting, Laplace smoothing shows the best performance. In the better performing USU sampling, instead the order is reversed. Here it is again Lidstone smoothing with a small setting for λ which provides the lowest deviation when estimating the distribution of data items over extended characteristic sets.

While in the previous examples we observed relatively small difference in quality between the individual smoothing techniques, we can see some examples for a stronger impact in Figure 5. In both shown plots we observe strong changes in Kullback-Leibler divergence. The most extreme case certainly is Figure 5(b), where the divergence reaches values of up to 3 at small sampling rates.



(a) TS, triple sampling (b) ECS, triple sampling

Figure 5. Examples for smoothing having a strong impact on Kullback-Leibler divergence.

6 Related Work

In the database world the estimation of densities has a long tradition in the context of query result set size estimation. Mannino et al. [16] describe how a statistical profile of a database can be used for query optimisation and performance prediction. The paper describes in detail how estimates for distributions can be used to estimate the cardinality of result sets for typical operators (join, select, etc.). Also aggregate operators (count, average) can benefit from statistical information over the data distribution [10]. A good and brief overview of sampling approaches in the context of database systems can be found in [15].

For sampling on RDF data there is relatively little work, so far. Sundra et al. [19] use samples of large graphs for the purpose of visualising data. The reduction in data volume helps to generate visualisations which can still be interpreted by human end users. Harth et al. [9] use data summaries to implement an approximative index structure over Linked Data. The data summaries correspond to initial random samples of data from the Web which are subsequently extended. Also the construction of graph summaries corresponds to a certain degree to sampling RDF data. Campinas et al. [2] used such graph summaries to assist users in formulating SPARQL queries. However, none of these works systematically investigated the impact of sampling methods on the quality of the resulting smaller data graph.

Estimates of distributions over RDF Data find their application in several scenarios. Obtained from locally stored data sets they are used for the purpose of query optimisation to estimate the result set size for query fragments [17]. The knowledge of these selectivity estimates is used for optimising execution plans of queries. Also in a federated setting of distributed data sources such result set size estimations have been employed [5]. Another application making use of distributions of the data are pattern detection methods such as an information theoretic analysis of LOD for the purpose of detecting redundancy on a schema level [7] or a statistical schema induction [20].

7 Conclusions

In this paper I addressed the task of sampling the LOD graph for the purpose of estimating data distributions. I looked at the effects of standard sampling and smoothing techniques on the quality of the estimates of distributions. In an empirical evaluation it could be observed that an USU based sampling over the modelled entities provides the best results for obtaining reliable distributions. This could motivate data providers to describe their entire data set with a small excerpt based on this sampling technique. The examples could easily be incorporated, for instance, in a VoID description [1] using void:exampleResource. Otherwise, the experiments showed that context based sampling—which is the most plausible and realistic sampling technique for LOD—provides acceptable results. However, it requires higher sampling rates. Regarding smoothing techniques it was not possible to identify a general best choice. In many cases the techniques had hardly any effect on the quality of the estimated distributions.

While the paper at hand gives an overview of how existing standard techniques perform, it can only provide first insights and a baseline for further investigations. Thus, in future work I will investigate alternative smoothing techniques. A very promising idea seems to adopt generalised approaches from the context of language modelling [18] and transfer them to the domain of Linked Open Data. The idea here is to break down unobserved composite structures (e.g. a type set) into groups of smaller composites which have been observed in a sample.

Acknowledgements The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013), REVEAL (Grant agree number 610928).

References

1. Alexander, K., Cyganiak, R., Hausenblas, M., Zhao, J.: Describing linked datasets with the void vocabulary. W3C Interest Group Note (Mar 2011), http://www.w3.org/TR/void/, (accessed 14 March 2014)
2. Campinas, S., Perry, T.E., Ceccarelli, D., Delbru, R., Tummarello, G.: Introducing rdf graph summary with application to assisted sparql formulation. In: Proceedings of the 23rd International Workshop on Database and Expert Systems Applications (2012)
3. Carothers, G.: Rdf 1.1 n-quads. W3C Recommendation (Feb 2014), http://www.w3.org/TR/2014/REC-n-quads-20140225/, (accessed 14 March 2014)
4. Dividino, R., Scherp, A., Gröner, G., Gottron, T.: Change-a-LOD: Does the Schema on the Linked Data Cloud Change or Not? In: COLD'13: International Workshop on Consuming Linked Data (2013)
5. Görlitz, O., Staab, S.: Splendid: Sparql endpoint federation exploiting void descriptions. In: Proceedings of the 2nd International Workshop on Consuming Linked Data. Bonn, Germany (2011)
6. Gottron, T., Gottron, C.: Perplexity of index models over evolving linked data. In: ESWC'14: Proceedings of the Extended Semantic Web Conference (2014)
7. Gottron, T., Knauf, M., Scheglmann, S., Scherp, A.: A Systematic Investigation of Explicit and Implicit Schema Information on the Linked Open Data Cloud. In: ESWC'13: Proceedings of the 10th Extended Semantic Web Conference. pp. 228–242 (2013)

8. Gottron, T., Knauf, M., Scherp, A.: Analysis of schema structures in the linked open data graph based on unique subject uris, pay-level domains, and vocabulary usage. Distributed and Parallel Databases pp. 1–39 (2014), http://dx.doi.org/10.1007/s10619-014-7143-0
9. Harth, A., Hose, K., Karnstedt, M., Polleres, A., Sattler, K.U., Umbrich, J.: Data summaries for on-demand queries over linked data. In: Int. Conf. on World wide web. pp. 411–420. ACM (2010), http://doi.acm.org/10.1145/1772690.1772733
10. Hou, W.C., Ozsoyoglu, G.: Statistical estimators for aggregate relational algebra queries. Transactions on Database Systems 16(4), 600–654 (Dec 1991)
11. Huffman, D.A.: A method for the construction of minimum redundancy codes. Proceedings of the I.R.E. 40(9), 1098–1101 (1952)
12. Käfer, T., Abdelrahman, A., Umbrich, J., O'Byrne, P., Hogan, A.: Observing Linked Data Dynamics. In: The Semantic Web: Semantics and Big Data, 10th International Conference. pp. 213–227. ESWC 2013 (2013)
13. Käfer, T., Umbrich, J., Hogan, A., Polleres, A.: DyLDO: Towards a Dynamic Linked Data Observatory. In: Workshop on Linked Data on the Web (LDOW) (2012)
14. Konrath, M., Gottron, T., Staab, S., Scherp, A.: Schemex—efficient construction of a data catalogue by stream-based indexing of linked data. Web Semantics: Science, Services and Agents on the World Wide Web 16(0), 52 – 58 (2012), the Semantic Web Challenge 2011
15. Ling, Y., Sun, W.: A supplement to sampling-based methods for query size estimation in a database system. SIGMOD Record 21(4), 12–15 (1992)
16. Mannino, Micheal, V., Chu, P., Sager, T.: Statistical profile estimation in database systems. ACM Computing Surveys 20, 191–221 (1988)
17. Neumann, T., Moerkotte, G.: Characteristic sets: Accurate cardinality estimation for rdf queries with multiple joins. In: Proceedings of the 27th International Conference on Data Engineering, ICDE 2011, April 11-16, 2011, Hannover, Germany. pp. 984–994 (2011)
18. Pickhardt, R., Gottron, T., Körner, M., Wagner, P.G., Speicher, T., Staab, S.: A Generalized Language Model as the Combination of Skipped n-grams and Modified-Kneser Ney Smoothing. In: ACL'14: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (2014), (to appear)
19. Sundara, S., Atre, M., Kolovski, V., Das, S., Wu, Z., Chong, E.I., Srinivasan, J.: Visualizing large-scale rdf data using subsets, summaries, and sampling in oracle. In: Data Engineering (ICDE), 2010 IEEE 26th International Conference on. pp. 1048–1059 (March 2010)
20. Völker, J., Niepert, M.: Statistical schema induction. In: Proceedings of the 8th extended semantic web conference on The semantic web: research and applications - Volume Part I. pp. 124–138. ESWC'11, Springer-Verlag, Berlin, Heidelberg (2011), http://dl.acm.org/citation.cfm?id=2008892.2008907